

Software–Projekt 2 2013/2014

VAK 03-BA-901.02

Projektplan

Sebastian Bredehöft	sbrede@tzi.de	2751589
Patrick Damrow	damsen@tzi.de	2056170
Tobias Dellert	tode@tzi.de	2936941
Tim Ellhoff	tellhoff@tzi.de	2520913
Daniel Pupat	dpupat@tzi.de	2703053
Mohamadreza (Amir) Khostevan	amirkh@tzi.de	1234567

Inhaltsverzeichnis

1	Einleitung (Patrick)	4
1.1	Projektübersicht	4
1.1.1	Ziele	4
1.1.2	Hauptarbeitsaktivitäten und –produkte	5
1.1.3	Haupt-Meilensteine und grober Zeitplan	6
1.1.4	Benötigte Ressourcen	7
1.1.5	Budget	7
1.1.6	Kontaktdaten des Kunden	8
1.1.7	Mitarbeiter	8
1.2	Auszuliefernde Produkte	9
1.3	Evolution des Plans	9
1.4	Referenzen	9
1.5	Definitionen und Akronyme	10
2	Projektorganisation (Patrick)	11
2.1	Prozessmodell	11
2.2	Organisationsstruktur	11
2.3	Organisationsgrenzen und –schnittstellen	12
2.4	Verantwortlichkeiten	12
3	Managementprozess (Daniel)	13
3.1	Managementprozess und -prioritäten	13
3.2	Annahmen, Abhängigkeiten und Einschränkungen	14
3.2.1	Annahmen	14
3.2.2	Abhängigkeiten	15
3.2.3	Einschränkungen	15
3.3	Risikomanagement	16
3.4	Projektüberwachung	21
3.5	Mitarbeiter	22
4	Technische Prozesse (Sebastian)	22
4.1	Methoden, Werkzeuge und Techniken	22
4.1.1	Entwicklungsplattform	22
4.1.2	Entwicklungsmethode	23
4.1.3	Programmiersprache und Bibliotheken	23
4.2	Dokumentationsplan	23
4.2.1	Codingstyle	23
4.2.2	Kommentarsprache	23
4.2.3	JavaDoc	23

4.2.4	Begleitende Dokumentation	24
4.3	Unterstützende Projektfunktionen	24
5	Arbeitspakete, Zeitplan und Budget (Tobias u. Tim)	24
5.0.1	Annahmen	24
5.0.2	Anmerkungen	24
5.1	Arbeitspakete, Zeitplan, Abhängigkeiten u. Ressourcenanforderungen . .	25
5.1.1	Projektplan	25
5.1.2	Anforderungsspezifikation	31
5.1.3	Architekturbeschreibung	34
5.1.4	Implementierung	34
5.1.5	Test	34
5.1.6	Dokumentabgaben	35
5.1.7	Sonstiges	35
5.1.8	Meetings	35
5.1.9	Kritischer Pfad	35
6	Sonstige Elemente (Amir)	35
6.1	Pläne für die Konvertierung von Daten	35
6.2	Managementpläne für Unterauftragsnehmer	35
6.3	Ausbildungspläne	35
6.4	Raumpläne	35
6.5	Installationspläne	35
6.6	Pläne für die Übergabe des Systems	35

Version und Änderungsgeschichte

Version	Datum	Änderungen
0.1	14.10.2013	Ziele hinzugefügt.
0.1.1	14.10.2013	Ziele vervollständigt
0.1.2	15.10.2013	Hauptarbeitsaktivitäten und -produkte hinzugefügt.
0.1.3	15.10.2013	Meilensteine eingefügt.
0.1.4	15.10.2013	Benötigte Ressourcen -Menschen hinzugefügt.
0.1.5	16.10.2013	Ressourcen ergänzt -Hardware und -Räume.
0.1.6	16.10.2013	Budget, Kontaktdaten und Mitarbeiter in den Projektplan eingefügt.
0.2	16.10.2013	Produkte, Evolution d. Plans, Defi. und Akronyme hinzugefügt.
0.2.1	17.10.2013	Prozessmodell und Organisationsstruktur eingefügt.
0.2.2	17.10.2013	Org.grenzen und -schnittst., Verantwortlichkeiten eingefügt.
0.2.3	17.10.2013	Tabellen in 1 und 2 eingefügt.
0.3	19.10.2013	Managementprozesse hinzugefügt.
1.0	20.10.2013	Erste veröffentlichte Version.

1 Einleitung (Patrick)

Dieses Dokument ist der Projektplan der Gruppe IT_R3V0LUT10N im des Softwareprojekts im Wintersemester 2013/2014 an der Universität Bremen. Der Projektplan entspricht der Struktur ANSI/IEEE Std. 1058.1-1987¹.

1.1 Projektübersicht

1.1.1 Ziele

Das Ziel unserer Gruppe IT_R3V0LUT10N ist es, das Softwareprojekt 2 der Universität Bremen zu bestehen. Dies setzt die Einhaltung der Fristen und Termine, eine aus-reichende Fertigstellung des Projekts und die Abgabe aller in SWP2 geforderten Dokumente wie Projektplan, Anforderungsspezifikation und Angebot, Architekturbeschreibung, Schnittstellenbeschreibung, Testplan inklusive Blackbox-Tests und ein elektronisch geführtes Berichtsheft voraus. Darüber hinaus wollen wir einen GUI-Prototypen erstellen und den Akzeptanztest bestehen. Ein Bibliothekssystem zu erstellen steht aber im Vordergrund.

Das Bibliothekssystem beinhaltet sowohl eine Website, als auch einen Zugang für mobile Geräte mit kleinem Display. Ziel ist es, die Mindestanforderungen² und eventuell weitergehende Funktionen zu implementieren.

¹<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=25325&userType=inst>

²http://www.informatik.uni-bremen.de/st/Lehre/swpII_1314/mindestanforderungen.html

Zu den Mindestanforderungen gehören die Erstellung und Abgabe einer Bibliothekssoftware, eines Serverprogramms mit Datenbankanbindung, einen Administrationszugang und einen Zugang für mobile Geräte mit kleinem Display. Wir haben uns entschieden den Zugang für die mobilen Geräte in Form einer Android-App zu realisieren, weil wir dies für Zeitgemäß und einfacher für den Leser, an den diese Form des Zugangs gerichtet ist, halten. Die zu erstellende Bibliothekssoftware dient in erster Linie zur Verwaltung des Medienbestandes der Bibliothek und dem Verleihen dieser Medien an der Oberschule Rockwinkel. Der Administrationszugang wird benötigt um Bibliothekare anzulegen, zu löschen, deren Stammdaten zu ändern, oder deren Rechte innerhalb der Software zu verändern.

(TODO: Weitere Features sind den oben genannten Mindestanforderungen zu entnehmen, oder werden im weiteren Verlauf des Projektplans erläutert.)

1.1.2 Hauptarbeitsaktivitäten und –produkte

In einem Softwareprojekt wird der Entwicklungsprozess einer Software in verschiedene Phasen unterteilt.

Da das Projekt aus verschiedenen Aktivitäten besteht, lassen sich diese Aktivitäten zu Arbeitsprodukten zusammenfügen. Die Folgende Tabelle bietet eine Übersicht der einzelnen Aktivitäten und den daraus resultierenden Arbeitsprodukten. Die Arbeitsprodukte werden im Laufe des Projektes nach und nach abgegeben.

Tabelle 1: Hauptaktivitäten und –produkte

Aktivität / Phase	Arbeitsprodukt
Projektplanung	Projektplan
Anforderungsanalyse, Angebotserstellung	Anforderungsspezifikation, Angebot
Entwurf (Globale Analyse, Konzeptionelles Modell, Modulperspektive, Ausführungsperspektive, Codeperspektive)	Architekturbeschreibung
erstellen des Testplans, Tests	Testplan, Schnittstellentests
Implementierung	lauffähiges Programm
Dokumentation	Installationsanweisung/-Skript
Auslieferung	Kunde erhält Produkt

1.1.3 Haupt-Meilensteine und grober Zeitplan

Die Haupt-Meilensteine resultieren aus den jeweiligen Abgabeterminen der einzelnen Dokumente. Weitere Meilensteine ergeben sich aus den Terminen die auf der Homepage³ des Moduls Software Projekt 2 der Universität Bremen vermerkt sind. Intern haben wir die Fertigstellung der einzelnen Abgaben etwas vorgezogen, damit wir Pufferzeit aufbauen können, um die Abgaben gegebenenfalls zu korrigieren wenn nötig.

M0 - 11.10.2013 Kick-Off Veranstaltung.

M1 - 14.10.2013 Beginn des Projektes.

M2 - 20.10.2013 Abgabe initialer Projektplan.

Jedes Mitglied muss seinen Teil fertig gestellt haben. Anschließend werden alle Einzelteile zusammengeführt und von allen auf Korrektheit geprüft.

M3 - 23.10.2013 Kundengespräch in der Oberschule Rockwinkel.

M4 - 6-7.11.2013 Vorstellen des Prototypen in der Oberschule Rockwinkel.

M5 - 13.11.2013 Anforderungsspezifikation (Intern).

Jedes Mitglied hat seinen Teil der Anforderungsspezifikation fertiggestellt. Anschließend werden die Teile zusammengeführt und von allen auf Korrektheit geprüft.

M6 - 17.11.2013 Abgabe der Anforderungsspezifikation, GUI-Protoyp und Angebot. Meilenstein 5 muss bereits fertig sein. Der GUI-Prototyp muss vollständig entwickelt sein. Abgabe via MEMS.

M7 - 18.12.2013 Architektur- und Schnittstellenbeschreibung, Testplan, Tests (Intern).

Jedes Mitglied muss seine Aufgaben erfüllt haben. Teile werden zusammengeführt und kontrolliert. Tests müssen implementiert sein.

M8 - 22.12.2013 Architekturbeschreibung, Testplan und Schnittstellentests fertig. Meilenstein 7 muss bereits erreicht worden sein. Tests wurden lauffähig implementiert. Abgabe via MEMS.

M9 - 26.01.2014 Erste lauffähige Basisversion.

Jedes Mitglied muss seine Arbeitspakete fertig gestellt haben. Das Team muss die oben genannten Mindestanforderungen², die mit einem * markiert sind, implementiert haben.

M10 - 10.-15.02.2014 Akzeptanztest.

M11 - 23.02.2014 Vollständige Abgabe der Dokumente und der Software.

Die Software muss lauffähig und vollständig implementiert sein,
Abgabe des Build-/Installationsskriptes

³http://www.informatik.uni-bremen.de/st/Lehre/swpII_1314/abgaben.html

1.1.4 Benötigte Ressourcen

- **Menschliche Ressourcen**

An menschlichen Ressourcen stehen sechs Informatikstudenten der Universität Bremen zur Verfügung. Wir haben als durchschnittliche Arbeitszeit pro Woche und Person einen Aufwand von ca. 14,5 Stunden für das Projekt errechnet. Dieser Wert ergibt sich folgendermaßen:

Für das Modul Software Projekt 2 gibt es 9CP. 1CP entspricht 30 Semesterwochenstunden. $9 \times 30 = 270$ Stunden. Da wir 19 Wochen lang an dem Projekt arbeiten werden, ergibt sich ein aufgerundeter Wert von 14,5 Stunden pro Woche ($270 / 19 = 14,21$). Unsere Kontaktdaten sind dem Punkt Mitarbeiter zu entnehmen. **(TODO: Referenz)**

- **Hard-/ und Software**

Jedes unserer Mitglieder ist im Besitz, oder hat Zugriff, auf Computer, die folgenden Anforderungen und Verfügbarkeiten gerecht werden müssen:

- zum Anfertigen der Dokumente wird ein Textsatzprogramm benötigt (L^AT_EX wird bevorzugt).
- für die Entwicklung der Software müssen Java-Runtime, ein Texteditor und eine Entwicklungsumgebung mit Android-SDK installiert sein.
- Git wird zum gleichzeitigen Bearbeiten der Dokumente und zum Datenaustausch der Entwickler benötigt.

- **Räume**

Das Team wird sich während der gesamten Projektlaufzeit montags, soweit verfügbar, in einer der Lerninseln im GW2 A2370 oder A3440 der Universität Bremen von 10 Uhr bis 14 Uhr treffen. Weitere spezielle Räumlichkeiten werden nicht benötigt, da wir den Kontakt regelmäßig via Skype oder E-Mail gewährleisten.

1.1.5 Budget

Ein Budget für dieses Projekt in Form von Geld entfällt, da die Software im Rahmen des Moduls Software Projekt 2 entwickelt wird. Wenn wir über 19 Wochen (vom 14.10.2013 bis zum 23.02.2014) an dem Projekt mit 6 Studenten 14,5 Stunden pro Woche arbeiten, ergibt sich eine Gesamtsumme von 1653 Entwicklerstunden ($19 \times 6 \times 14,5 = 1653$).

Wir entnehmen einer Studie von Gulp⁴ das zwei Drittel der Software-Entwickler zwischen 60 und 80 Euro fordern. Da wir alle Studenten sind und somit noch in der Ausbildung, setzen wir den Stundenlohn für jeden Entwickler bei 40 Euro an. Somit würden sich für den Arbeitsaufwand der Entwicklerstunden Kosten von insgesamt 66.120 Euro ergeben.

⁴<http://www.gulp.de/presse/pressemitteilungen/marktstudie-freiberufliche-software-entwickler-sind.html>

1.1.6 Kontaktdaten des Kunden

Auftraggeber:

Herr Meyhöfer

Öffentliche Schulbibliothek Rockwinkel

in der Oberschule Rockwinkel

Uppe Angst 31, 28355 Bremen

Telefon : +49-(0)421 / 361-3360

Übergeordnete Organisation:

Prof. Dr. rer. nat. Rainer Koschke

OAS, Linzer Str. 9a

28359 Bremen

Telefon : +49-(0)421 / 218-2421

Telefon : +49-(0)421 / 218-9671

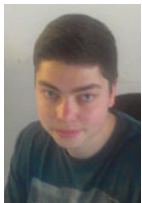
Fax : +49-(0)421 / 218-4322

E-Mail: koschke@informatik.uni-bremen.de

1.1.7 Mitarbeiter

In der Folgenden Tabelle 2 stehen die Kontaktdaten aller am Projekt Beteiligten. Ihr sind in Folge Nachname, Name, E-Mail und ein Foto des jeweiligen Teammitglieds zu entnehmen.

Tabelle 2: Mitarbeiter

Name	Email	Foto
Bredehöft, Sebastian	sbrede@tzi.de	bild einfügen
Damrow, Patrick	damsen@tzi.de	bild einfügen
Dellert, Tobias	tode@tzi.de	bild einfügen
Pupat, Daniel	dpupat@informatik.uni-bremen.de	
Ellhoff, Tim	tellhoff@tzi.de	bild einfügen
Khostevan, Mohamadreza (Amir)	amirkh@tzi.de	bild einfügen

1.2 Auszuliefernde Produkte

Die Tabelle 3 listet alle auszuliefernde Produkte auf die während des gesamten Projektes anfallen werden.

Tabelle 3: Auszuliefernde Produkte

Datum	Beschreibung	Anzahl
20.10.2013	Initialer Projektplan (dieses Dokument)	1
23.10.2013	aktuelles Berichtsheft (von jedem Mitglied)	6
17.11.2013	Anforderungsspezifikation	1
17.11.2013	GUI-Prototyp	1
17.11.2013	Angebot	1
20.11.2013	aktuelles Berichtsheft (von jedem Mitglied)	6
22.12.2013	Architekturbeschreibung	1
22.12.2013	Schnittstellenbeschreibung	1
22.12.2013	Testplan	1
25.12.2013	aktuelles Berichtsheft (von jedem Mitglied)	6
26.01.2014	Erste lauffähige Basisversion	1
23.02.2014	Vollständige Abgabe	1
23.02.2014	unterschiedene Erklärung zum Berichtsheft	1

1.3 Evolution des Plans

Der Projektplan wird über die gesamte Dauer der Entwicklung durch das Team aktualisiert. Die erste absehbare Aktualisierung wird nach dem Kundengespräch am 23.10.2013, 9 Uhr, in der Oberschule Rockwinkel, durchgeführt. Weitere absehbare Aktualisierungen des Projektplans sind nach den jeweiligen Hauptabgaben vom Entwicklerteam durchzuführen.

Aufgrund der stetigen Entwicklung des Systems sind weitere Aktualisierungen abzusehen. Vor allem im Abschnitt Arbeitspakete (**TODO: ref Abschnitt Arbeitspakete**) des Projektplans. Alle Änderungen werden von dem jeweiligen Phasenleiter (**TODO: ref Phasenl.**) überwacht und Korrektheit geprüft. Als unvorhergesehen Aktualisierungen wären z.B. das Austreten eines Mitglieds aus der Gruppe zu nennen, da dies die meiste Umstrukturierung mit sich zieht. Die Arbeitspakete müssten in dem Fall neu auf die restlichen Teammitglieder aufgeteilt werden, was wiederum der jeweilige Phasenleiter übernimmt.

1.4 Referenzen

1.5 Definitionen und Akronyme

Tabelle 4: Definition und Akronyme

Begriff	Bedeutung
Java	Java ist eine Programmiersprache
Eclipse	die von uns genutzte Entwicklungsumgebung
Git/GitHub	Ein Online-Projekt Hoster für die Verwaltung und Versionierung von Dokumenten und Sourcecode
GUI	Grafische Oberfläche, Abkürzung für Graphical User Interface
Ansi/IEEE	eine festgelegte Norm vom 'Institute of Electrical and Electronics Engineers, ANSI ist die Abkürzung für 'American National Standards Institute'
Wasserfallmodell	Ein lineares Vorgehensmodell der Softwareentwicklung
L ^A T _E X	Ein Textsatzprogramm
MEMS	Ein elektronisches Abgabesystem der Universität Bremen
Build-/Installationsskript	Anleitung zum Installieren der Software
CP	Credit Points, 1CP entspricht 30 Semesterwochenstunden Arbeitsaufwand
Server	Ein dauerhaft erreichbarer Rechner, der einen Dienst bereitstellt
Android-SDK	SDK = Software-Development-Tool
Skype	Online Kommunikations-Tool
E-Mail	Elektronischer Brief

2 Projektorganisation (Patrick)

2.1 Prozessmodell

Das von uns verwendete Wasserfallmodell gliedert die einzelnen Phasen sequentiell aufeinanderfolgend:

- Anforderungsspezifikation
- Architekturbeschreibung (Entwurf)
- Implementierung
- Test
- Dokumentation
- Auslieferung

In unserem Fall werden sich die Phasen Implementierung, Test und Dokumentation überschneiden um eine effiziente Arbeitsweise zu gewährleisten.

2.2 Organisationsstruktur

In unserer Organisationsstruktur ist es vorgesehen, dass es für jede Phase des Projekts einen Phasenleiter gibt. Da es keinen Projektleiter gibt, übernimmt der Phasenleiter dessen Funktionen in der jeweiligen Phase. Die einzelnen Phasenleiter werden in Abstimmung aller Mitglieder gewählt. Somit übernimmt jedes Mitglied einmal die Rolle des Phasenleiters und die Verantwortung, die in der entsprechenden Phase anfallenden Arbeitspakete auf die restlichen Mitglieder zu verteilen, erledigte Arbeitspakete auf Korrektheit zu überprüfen, die einzelnen Termine einzuhalten und die Qualität des jeweiligen Phasenprodukts zu gewährleisten.

Die anderen Teammitglieder haben den Anweisungen des Phasenleiters zu folgen und müssen alle erteilten Aufgaben fristgerecht bearbeiten und beim Phasenleiter einreichen. Falls Probleme erkenntlich werden, sind diese dem Phasenleiter sofort mitzuteilen, damit auf die Ereignisse reagiert werden kann.

Im Rahmen des Projekts muss jedes Mitglied regelmäßig seine E-Mails kontrollieren. Für eine vereinfachte Kommunikation haben wir zusätzlich alle Kontaktdaten untereinander ausgetauscht, welche Telefonnummern, E-Mailadressen und Skype-ID's beinhalten. Einen E-Mailverteiler wird auch in absehbarer Zeit eingerichtet und an entsprechender Stelle ergänzt.

Regelmäßige Treffen sind einmal wöchentlich, am Montag, im GW2 A3440 Lerninsel 3 der Universität Bremen (soweit verfügbar) ab 10 Uhr angesetzt. Hier sollen Arbeitspakete, Fortschritt und Probleme ausgetauscht und besprochen werden.

Um zu gewährleisten das alle Teammitglieder die aktuelle Version der Dokumente zur

Bearbeitung besitzen, haben wir ein Repository bei GitHub eingerichtet, welches jedes Mitglied regelmäßig zu aktualisieren hat.

2.3 Organisationsgrenzen und –schnittstellen

Bei dem Arbeitgeber und der übergeordneten Organisation handelt es sich um zwei verschiedene Parteien, da es sich bei dem Auftraggeber um einen echten Kunden handelt.

Auftraggeber:

Herr Meyhöfer

Öffentliche Schulbibliothek Rockwinkel

in der Oberschule Rockwinkel

Uppe Angst 31, 28355 Bremen

Telefon : +49-(0)421 / 361-3360

Übergeordnete Organisation:

Prof. Dr. rer. nat. Rainer Koschke

OAS, Linzer Str. 9a

28359 Bremen

Telefon : +49-(0)421 / 218-2421

Telefon : +49-(0)421 / 218-9671

Fax : +49-(0)421 / 218-4322

E-Mail: koschke@informatik.uni-bremen.de

2.4 Verantwortlichkeiten

Wie in Abschnitt 2.2 bereits erwähnt wechselt bei jeder Phase der Phasenleiter, welcher während der jeweiligen Phase ebenfalls für das Projektmanagement zur Verantwortung zu ziehen ist. In Tabelle 5 werden alle Phasen mit dem zugehörigen Phasenleiter aufgelistet.

Tabelle 5: Verantwortlichkeiten

Phase	Verantwortlicher
Projektplan	Patrick Damrow
Anforderungsspezifikation und Angebot	Daniel Pupat
Architekturbeschreibung und Entwurf	Tim Ellhoff
Implementierung	Tobias Dellert
Test	Sebastian Bredehöft
Dokumentation	Mohamadreza (Amir) Khostevan

Weitere Verantwortlichkeiten, die von jeweils einem Teammitglied während des gesamten Projektzeitraumes besetzt werden müssen, sind Qualitätsmanagement und Risikoman-

gement. Die Aufgaben des Qualitätsmanagers sind es, die Qualität aller Bearbeitungen der Teammitglieder sicherzustellen. Zu den Aufgaben des Risikomanagers gehören das frühzeitige Erkennen von möglichen Problemen und diese präventiv zu vermeiden oder einzudämmen, indem er z.B. Arbeitspakete umverteilt oder andere Lösungen findet. siehe Abschnitt **(TODO: ref auf Risikomanagement)**

3 Managementprozess (Daniel)

3.1 Managementprozess und -prioritäten

Folgende Managementprozesse haben bei uns die höchsten Prioritäten:

Fertigstellung des Produktes:

Ein Ziel von uns ist die Fertigstellung des Produktes, welches vom Kunden gefordert ist. Dabei ist wichtig, dass alle Mindestanforderungen, die der Kunde gefordert hat, erfolgreich umgesetzt wurden.

Dieses Ziel hat die höchste Priorität, da dies notwendig ist, um die Veranstaltung SWP2 zu bestehen und eine Voraussetzung aller anderen aufgeführten Ziele ist.

Qualität des Produktes:

Ein weiteres Ziel ist es, dem Produkt eine hohe Qualität zu geben. Dies ist notwendig, damit der Kunde zufrieden ist und das Produkt später evtl. verwendet wird. Dabei ist wichtig, dass neben den Mindestanforderungen weitere Funktionen vorhanden sind und die Benutzung einfach und benutzerfreundlich ist.

Dieses Ziel hat eine hohe Priorität, da dies notwendig ist um den Kunden zufriedenzustellen und eine gute Note zu erreichen.

Weiterentwicklung des Produktes:

Es ist auch wichtig, das Produkt so zu entwickeln, damit dieses später bei Bedarf von Anderen weiterentwickelt werden kann. Dies erfordert eine strukturierte Implementierung.

Dieses Ziel hat niedrige Priorität, da wir in erster Linie das Modul bestehen wollen.

Kundenzufriedenheit:

Es ist sehr wichtig, dass der Kunde später zufrieden ist, was bedeutet, dass man die Mindestanforderungen erfüllt und darüber hinaus noch weitere Features einbindet, da nur so der Kunde wirklich zufrieden ist.

Dieses Ziel hat für uns mittlere Priorität, da wir in erster Linie die Mindestanforderungen schaffen wollen und nur wenn noch Zeit ist, weitere Features einbinden. Dies könnte aber noch notwendig sein, um eine gute Note zu erreichen.

Kommunikation innerhalb der Gruppe:

Ein wichtiger Faktor ist die Kommunikation innerhalb der Gruppe. Wenn man sich nicht abspricht, kann es zu Schwierigkeiten kommen, wenn z.B. ein Gruppenmitglied seinen Teil nicht rechtzeitig schafft und die anderen aber davon ausgehen.

Dieses Ziel hat bei uns eine hohe Priorität, da ohne Kommunikation das Projekt mit hoher Wahrscheinlichkeit scheitert.

Klima innerhalb der Gruppe:

Ein gutes Gruppenklima heißt, dass innerhalb der Gruppe alle gut miteinander auskommen und es keinen Streit gibt. Außerdem muss man den Anderen vertrauen können, dass sie immer rechtzeitig fertig werden und bei Problemen Bescheid geben.

Dies hat ebenfalls eine hohe Priorität, da gerade das Vertrauen und die Zuverlässigkeit sehr wichtig sind, damit alles rechtzeitig fertig wird.

Gute Note:

Ziel dieser Veranstaltung ist für uns das Projekt so gut wie möglich zu bestehen. Dabei sollte jeder sein Bestes geben, damit am Ende das Maximum an Punkten für die Gruppe erreicht wird.

Dies hat bei uns eine hohe Priorität, da wir später einen möglichst guten Abschluss haben wollen.

Kunde entscheidet sich für unser Produkt:

Da der Kunde am Ende der Veranstaltung ein Produkt aussuchen wird, welches dann in der Bibliothek verwendet wird, wäre es möglich, dass er unser Produkt wählt.

Dieses Ziel hat bei uns eine niedrige Priorität, da wir in erster Linie gut abschneiden wollen, aber nicht darauf hinarbeiten, unbedingt das beste Produkt der Veranstaltung zu entwickeln, da dies zu zeitaufwendig wäre.

3.2 Annahmen, Abhängigkeiten und Einschränkungen

3.2.1 Annahmen

Mindestanforderungen werden nicht verändert:

Die erste Annahme ist, dass der Kunde die Mindestanforderungen nicht verändert. Dies bedeutet, dass es keine Möglichkeit gibt, andere Mindestanforderungen auszuhandeln und der Kunde auch keine neuen stellt.

Deadline wird nicht verschoben:

Noch eine Annahme ist, dass sich die Deadlines der verschiedenen Abgaben unter normalen Umständen nicht verändern. Dies bedeutet, dass der Kunde diese nicht vorverlegt und wir diese nicht nach hinten verlegen können.

Erfolgreiche Teilnahme:

Eine weitere Annahme ist, dass alle Gruppenmitglieder die Veranstaltung erfolgreich bestehen wollen. Da sich alle für dieses Modul eingetragen haben, kann man davon ausgehen, dass alle ihr Bestes geben, um diese Veranstaltung zu bestehen.

Grundkenntnisse in Java:

Man kann auch annehmen, dass alle Mitglieder Grundkenntnisse in Java haben, da alle Gruppenmitglieder bereits die Veranstaltungen Praktische Informatik 1 und 2 besucht haben.

3.2.2 Abhängigkeiten

Laptop:

Da jeder von uns ein Laptop besitzt, werden wir diesen hauptsächlich verwenden, da so jeder mobil ist und überall weiterarbeiten kann.

GitHub:

Zum Teilen der Dokumente verwenden wir GitHub. So kann jede Person einen Teil bearbeiten und die Dokumente können dann zusammengeführt werden.

Glassfish:

Als Server verwenden wir den Glassfish Server, auf dem unser Programm später arbeitet.

Mitglieder:

Da dies eine Gruppenarbeit ist, muss jedes Gruppenmitglied seinen Teil leisten, da die Arbeit auf sechs Leute ausgelegt ist.

Von den eben genannten Punkten ist das Projekt abhängig, da bei einem Ausfall der Punkte Schwierigkeiten auftreten können.

3.2.3 Einschränkungen

Weitere belegte Module:

Jeder von uns belegt noch weitere Module und hat deswegen nur eine gewisse Zeit für SWP2. Hinzu kommt noch, dass wir Mitglieder haben, die in unterschiedlichen Semestern sind, wodurch es schwierig ist, einen gemeinsamen Termin zu finden.

3.3 Risikomanagement

Risiko	EW (1-10)	SH (1-10)	RH
Krankheitsbedingter Ausfall eines Gruppenmitglieds	5	4	20
Krankheitsbedingter Ausfall mehrerer Gruppenmitglieder	2	7	14
Austritt eines Gruppenmitglieds	4	5	20
Austritt mehrerer Gruppenmitglieder	1	8	8
Inkompetenz eines Gruppenmitglieds	2	7	14
Mangelhafte Kommunikation innerhalb der Gruppe	4	6	24
Auflösung/Teilung der Gruppe	2	10	20
Unstimmigkeiten in der Gruppe	2	5	10
Mangelnde Motivation in der Gruppe	6	5	30
Zeitmangel	6	6	36
Probleme mit neuen Technologien	5	3	15
Ausfall von GitHub	1	8	8
Ausfall des Glassfish-servers	1	8	8
Mindestanforderung des Kunden falsch interpretiert	3	9	27
Fehler in Architekturbeschreibung	3	7	21

EW = Eintrittswahrscheinlichkeit (Skala 1:gering - 10:hoch)

SH = Schadenshöhe (Skala 1:gering - 10:hoch)

RH = Risikohöhe (EW * SH)

Krankheitsbedingter Ausfall eines/mehrerer Gruppenmitglieds/er:

Aufgrund von Krankheiten fallen eine oder mehrere Personen aus und können nicht mehr richtig oder für eine gewisse Zeit überhaupt nicht mehr mitarbeiten. Dadurch kommt auf die restliche Gruppe mehr Arbeit zu.

Maßnahmen zur Schadensbegrenzung:

1. Gruppenmitglied benachrichtigt die anderen Mitglieder so früh wie möglich, damit diese sich darauf einstellen können.
2. Die Gruppe sucht Gespräch mit dem Tutor, wenn mehrere Personen ausfallen.

Austritt eines/mehrerer Gruppenmitglieds/er:

Aufgrund von Zeitmangel, Studienabbruch und anderen Gründen kann es jederzeit passieren, dass Gruppenmitglieder aus der Gruppe austreten. Dadurch müssen die anderen Personen dann entsprechend mehr arbeiten, was zu Problemen führen kann.

Maßnahmen zur Vorbeugung:

1. Im Zeitplan vor den Deadlines immer ein wenig Zeit überlassen, um durch einen plötzlichen Austritt die Abgabe noch rechtzeitig zu schaffen.

Maßnahmen zur Schadensbegrenzung:

2. Bei einem Austritt aus der Gruppe gibt das Mitglied den anderen sofort Bescheid, damit diese sich rechtzeitig auf die Mehrarbeit einstellen können.
3. Sollten mehrere Mitglieder austreten, Gespräch mit dem Tutor suchen um gegebenenfalls die Anforderungen zu senken.

Inkompetenz eines Gruppenmitglieds:

Es kann passieren das ein Gruppenmitglied Inkompetent ist und somit nicht in der Lage die ihm zugeteilten Aufgaben zu lösen. Das kann daran liegen, dass dieses Mitglied im bisherigen Studienverlauf immer durch andere Gruppenmitglieder die Module bestanden hat. Dabei kann es passieren, dass dieses Unentdeckt bleibt und so erst spät festgestellt wird das ein Teil nicht funktioniert.

Maßnahmen zur Schadensbegrenzung:

1. Durch wöchentliche Treffen wird der Fortschritt besprochen und wenn jemand dieses nicht hinbekommt, wird dieses frühzeitig erkannt und der Teil kann unter den anderen Mitgliedern aufgeteilt werden.
2. Jedes Gruppenmitglied sollte sich bei Problemen frühzeitig an die Gruppe wenden, damit die anderen diesen helfen können.
3. Wenn dieses Gruppenmitglied zu einer Belastung für die Gruppe wird, kann durch eine Gruppenentscheidung dieses Mitglied aus der Gruppe ausgeschlossen werden.

Mangelhafte Kommunikation innerhalb der Gruppe:

Da wir in einer relativ großen Gruppe arbeiten, wird die Arbeit aufgeteilt. Dabei sind viele Teilaufgaben abhängig von anderen, was dazu führen kann, dass ohne Kommunikation die Teilaufgaben oder Implementierungen nicht zusammenpassen. Dadurch kann es im späteren Verlauf zu großen Problemen kommen, da so die Software evtl. nicht läuft.

Maßnahmen zur Vorbeugung:

1. Die Mitglieder, die stark voneinander abhängige Aufgaben haben, sollte sich vorher genau absprechen und auch die ganze Zeit in Kontakt bleiben, um Missverständnisse zu vermeiden.
2. In den wöchentlichen Treffen jede Aufgabe ansprechen, damit jeder weiß, was ungefähr die anderen Mitglieder wie machen.

Maßnahmen zur Schadensbegrenzung:

3. Jeder sollte sich die Aufgaben der anderen Mitglieder immer durchlesen und gerade bei abhängigen Aufgabenteilen genau drauf achten, dass diese zusammenpassen.

Teilung der Gruppe:

Wenn es Probleme oder Unstimmigkeiten innerhalb der Gruppe gibt und sich zwei Lager bilden, kann es dazu führen, dass sich die Gruppe trennen muss. Dies kann auch passieren, wenn sich herausstellen sollte, dass die Mitglieder starke unterschiedliche

Fähigkeiten besitzen und so die besseren Personen die Hauptarbeit verrichten müssen und diese damit nicht einverstanden sind.

Maßnahmen zur Vorbeugung:

1. Bei unterschiedlichen Fähigkeiten früh festlegen, dass diese später bessere Noten bekommen, damit diese auch zufrieden sind.

Maßnahmen zur Schadensbegrenzung:

2. Bei Problemen und Unstimmigkeiten das Gespräch suchen und diese Ansprechen und gegebenenfalls auch den Tutor hinzuziehen, um die Probleme zu lösen.

Auflösung der Gruppe:

Es kann durch schlechte Abgaben, Zeitmangel, Unstimmigkeiten und Gruppenaustritten eine Gruppenauflösung geben, was zu einem nicht bestehen des Moduls führen würde, da man alleine dieses wahrscheinlich nicht hinbekommen würde.

Maßnahmen zur Vorbeugung: 1. Bei Problemen frühzeitig das Gespräch suchen, um zu verhindern, dass sich die Gruppe auflöst.

Unzuverlässigkeit eines Gruppenmitglieds:

Wenn ein Gruppenmitglied seine Aufgaben nicht zu dem geplanten Zeitpunkt fertig bekommt, da er keine Zeit hatte oder andere Prioritäten gesetzt hat, kann es dazu führen, dass die Abgabe nicht vollständig ist und es eine 5.0 für die Abgabe gibt.

Maßnahmen zur Vorbeugung:

1. Bei der Planung der Zeit immer ein wenig Luft lassen, damit man noch reagieren kann, falls ein Mitglied seine Aufgaben nicht gemacht hat.

Maßnahmen zur Schadensbegrenzung:

2. Im wöchentlichen Treffen den Fortschritt jedes Mitglieds begutachten und evtl. darauf reagieren, sollte ein Mitglied nicht im Zeitplan sein.

3. Mitglied, wenn es die Aufgaben nicht macht beim ersten Mal ermahnen und bei wiederholten Male aus der Gruppe ausschließen.

Unstimmigkeiten in der Gruppe:

Da sich in der Gruppe Leute befinden, die sich vorher nicht kannten, kann es passieren, dass sich Mitglieder mit anderen Mitgliedern nicht verstehen und so das Gruppenklima stören.

Maßnahmen zur Schadensbegrenzung:

1. Bei einem Problem, müssen die anderen Gruppenmitglieder schlichten und z.B die Probleme in einem Gruppentreffen ansprechen und lösen.

Mangelnde Motivation in der Gruppe:

Da SWP 2 sehr zeitaufwendig ist und sich Abgaben über mehrere Wochen erstrecken,

kann es gerade am Anfang einer neuen Abgabe zu Mangelnder Motivation kommen, da man denkt, dass man noch genug Zeit hat.

Maßnahmen zur Vorbeugung:

1. Treffen mit der Gruppe planen, ohne am Projekt zu arbeiten, um die Moral der Gruppe zu stärken und sich gegenseitig zu motivieren.

Zeitmangel:

Da es feste Deadlines für die einzelnen Abgaben gibt, kann es zu Zeitproblemen kommen. Auch durch Ausfälle von Mitgliedern oder schlechte Zeiteinplanung kann es zu zeitlichen Mangel kommen.

Maßnahmen zur Vorbeugung:

1. Bei der Planung immer einen Zeitpuffer lassen, damit man noch vor den Abgaben Luft hat falls es irgendwelche Probleme während der Bearbeitungsphase gibt.
2. Bei den wöchentlichen Treffen immer überprüfen, ob jeder im Zeitplan ist, um notfalls noch früh genug auf Zeitprobleme zu reagieren.

Maßnahmen zur Schadensbegrenzung:

3. Wenn ein Aufgabenteil in Zeitverzug kommt, diesen auf mehreren Mitgliedern aufteilen, damit dieser noch rechtzeitig fertig wird.

Probleme mit neuen Technologien:

Da wir mit Technologien arbeiten müssen, mit denen wir noch keine oder nur wenig Erfahrung haben wie z.B. Android, kann dies zu unerwarteten Problemen führen. Dadurch kann sich der zeitliche Aufwand stark erhöhen.

Maßnahmen zur Vorbeugung:

1. Im Zeitplan bereits Zeit zur Einarbeitung neuer Programme einplanen, damit es noch genug Zeit zur Bearbeitung gibt.
2. Vorher die Mitglieder so aufteilen, dass die Personen die bereits Erfahrung mit den jeweiligen Technologien haben, diese Aufgabenteile übernehmen.

Maßnahmen zur Schadensbegrenzung:

3. Falls ein Gruppenmitglied mit einer Technologie überhaupt nicht zurecht kommt, können die Aufgaben mit einem anderen Mitglied getauscht werden.

Ausfall von GitHub

Wir benutzen GitHub als Repository. Es kann passieren, dass es Probleme mit dem Server gibt, was bei uns zu organisatorischen Problemen führen kann. **Maßnahmen zur Vorbeugung:**

1. Jeder sollte regelmäßig Back-ups durchführen, damit man sichergehen kann, dass keine Daten bei einem Serverabsturz verloren gehen.

Maßnahmen zur Schadensbegrenzung:

2. Bei einem Server Ausfall werden dem Phasenleiter alle Dokumente per E-Mail

geschickt, damit dieser diese zusammenfügen und abschicken kann.

Ausfall des Glassfish-servers

Da wir mit Glassfish arbeiten, kann es passieren, dass dieser kurzzeitig ausfällt. Das würde dazu führen, dass man das Programm nicht ausführen und somit nicht vernünftig testen kann.

Maßnahmen:

Da es keine Ausweichmöglichkeit gibt, können hier keine Maßnahmen getroffen werden.

Mindestanforderung des Kunden falsch interpretiert:

Da es viele Forderungen des Kunden gibt, kann es sein, dass eine falsch verstanden oder vergessen wird und das zu einem nicht bestehen führen würde.

Maßnahmen zur Vorbeugung:

1. Jedes Mitglied sollte vor der Abgabe für sich alleine noch einmal alle Anforderungen überprüfen und bei zweifeln die Gruppe informieren.
2. Bei unklaren Forderungen im Kundengespräch bereits konkrete Fragen dazu stellen.

Fehler in Architekturbeschreibung:

Es kann passieren, dass man in der Architekturbeschreibung bereits Fehler eingebaut hat, die dann später bei der Implementierung entdeckt werden. Dadurch können große Probleme auftreten, gerade wenn man versucht 2 Code Stücke zusammenzufügen.

Maßnahmen zur Vorbeugung:

1. Bei der Architekturbeschreibung konzentriert arbeiten und sich Zeit lassen für Klassendiagramme etc., damit dort möglichst keine Fehler auftreten.

Maßnahmen zur Schadensbegrenzung:

2. Sollte jemanden ein Fehler auffallen, der mehrere Komponenten betrifft, sollte dieser möglichst schnell mit der ganzen Gruppe gelöst werden.

Die Maßnahmen zur Vorbeugung werden über die gesamte Zeit des Projektes angewendet, bei den Maßnahmen zur Schadensbegrenzung nach Eintritt entscheidet der Phasenleiter, welche Maßnahmen getroffen werden.

Aufgrund der getroffenen Maßnahmen verändern sich die Werte wie folgt:

Risiko	NEW (1-10)	NSH (1-10)	NRH
Krankheitsbedingter Ausfall eines Gruppenmitglieds	5	3	15
Krankheitsbedingter Ausfall mehrerer Gruppenmitglieder	2	5	10
Austritt eines Gruppenmitglieds	4	3	12
Austritt mehrerer Gruppenmitglieder	1	6	6
Inkompetenz eines Gruppenmitglieds	1	4	4
Mangelhafte Kommunikation innerhalb der Gruppe	3	5	15
Teilung der Gruppe	1	8	8
Auflösung der Gruppe	1	10	10
Unzuverlässigkeit eines Gruppenmitglieds	2	7	14
Unstimmigkeiten in der Gruppe	1	5	5
Mangelnde Motivation in der Gruppe	5	5	25
Zeitmangel	4	5	20
Probleme mit neuen Technologien	4	2	8
Ausfall von GitHub	1	6	6
Ausfall des Glassfish-servers	1	8	8
Mindestanforderung des Kunden falsch interpretiert	2	9	18
Fehler in Architekturbeschreibung	3	5	15

NEW = Neue Eintrittswahrscheinlichkeit (Skala 1:gering - 10:hoch)

NSH = Neue Schadenshöhe (Skala 1:gering - 10:hoch)

NRH = Neue Risikohöhe (NEW * NSH)

3.4 Projektüberwachung

Um das Projekt zu überwachen, wird jeden Montag ein Treffen stattfinden, wo überprüft wird, wie weit jeder ist. Später wenn die Aufgaben stark voneinander Abhängig sind, wird es noch ein Treffen am Freitag geben, um sich untereinander besser abzusprechen. Dies wird gerade ab der Architektur wichtig. Außerdem wird es einen permanenten Austausch über Skype geben, indem über Probleme und Anliegen diskutiert wird. Dieser läuft in erster Linie über einen Gruppenchat ab, sollte es aber einmal etwas konkreter werden, wird untereinander telefoniert.

Es wird auch für jede Phase einen Phasenleiter geben, der dafür zuständig ist, den Zeitplan im Auge zu behalten. Diesem Phasenleiter muss dann jedes Gruppenmitglied regelmäßig Bescheid geben, wie weit die Teilaufgabe bereits bearbeitet ist. Sollte es Probleme geben, ist die Aufgabe des Phasenleiters das Risiko einzuschätzen und Maßnahmen zu treffen, welche im Punkt Risikomanagement erläutert sind.

Bei größeren Problemen kann ein Treffen spontan einberufen werden oder es wird direkt bei Skype angesprochen und da versucht, dies zu lösen.

3.5 Mitarbeiter

Die Mitarbeiter sollten ausreichend Programmierkenntnisse in Java haben, damit diese später auch in der Lage sind die Bibliothek zu entwerfen. Auch sollten diese zumindest an SWP 1 teilgenommen haben, um zumindest die Grundkenntnisse zu haben, um die Aufgaben zu bearbeiten.

Die Mitarbeiter sollten auch in der Lage zu sein im Team zu arbeiten und sich eigenständig in Themengebiete einzuarbeiten. Bei Problemen sollten die Mitarbeiter sich direkt an die Gruppe wenden, damit diese frühzeitig gelöst werden. Außerdem sollten sie genügend Kenntnisse in Deutsch und Englisch haben, um sich in den Sprachen vernünftig auszudrücken. Die Mitarbeiter sollten zudem noch genügend Kenntnisse in Latex haben, sowie SQL Statements erstellen können.

4 Technische Prozesse (Sebastian)

4.1 Methoden, Werkzeuge und Techniken

4.1.1 Entwicklungsplattform

Folgende Werkzeuge werden im Entwicklungsprozess von uns benutzt:

- Eclipse⁵ ist unsere Entwicklungsumgebung (beinhaltet AndroidSDK⁶ für die Androidentwicklung)
- Maven⁷ ist unser Build-Management Tool
- GlassFish 3.1⁸ ist unser Application-Server
- jUnit⁹ ist unser Framework zum Testen
- GitHub¹⁰ zur Versionsverwaltung
- GanttProject¹¹ für Gantt-Diagramme
- L^AT_EX¹² zur Dokumentenerstellung

⁵<http://www.eclipse.org/>

⁶<https://developer.android.com/sdk/index.html>

⁷<http://maven.apache.org/>

⁸<http://glassfish.java.net/>

⁹<http://junit.org/>

¹⁰<http://github.com/>

¹¹<http://www.ganttproject.biz/>

¹²<http://www.latex-project.org/>

4.1.2 Entwicklungsmethode

Unsere Entwicklungsmethode ist das Wasserfallmodell, wie in 2.1 vorgestellt. Dabei wird nicht alles sequentiell abgearbeitet, sondern die Phasen Implementierung, Test und Dokumentation werden sich überschneiden.

4.1.3 Programmiersprache und Bibliotheken

Die Programmiersprache wird Java (mindestens Version 5) sein, auch im Hinblick das wir das Anroid SDK benutzen. Außerdem wird in geringen Umfang HTML und XML benutzt.

Ob und welche Bibliotheken noch genutzt werden, kann zu diesem Zeitpunkt(*Abgabe: 20.10.2013*) nicht gesagt werden.

Sobald wir Bibliotheken nutzen, wird dieser Punkt aktualisiert.

4.2 Dokumentationsplan

Wir werden als Ergebnis verschiedene Dokumentationen vorweisen können. Diese sind:

- Nutzerhandbuch
- Installationsanleitung
- Dokumentation des Quellcodes

4.2.1 Codingstyle

Unsere Implementierungen werden sich an die *Code Conventions for the Java Programming Language*¹³ halten.

Unsere .tex Dateien gehen aus der Vorlage für das SWP2-Projekt hervor. Es ist dafür zu sorgen, dass es jederzeit kompilierbar ist.

4.2.2 Kommentarsprache

Die Sprache in der unsere Kommentare verfasst sind, wird Deutsch sein. Dies verhindert mögliche Missverständnisse innerhalb unserer Gruppe.

4.2.3 JavaDoc

Wir benutzen JavaDoc zur Dokumentation unseres Quellcodes. Dieses lässt eine einfache Erstellung von HTML-Dokumentationsdateien zu. Zur zukünftigen Wartung wird bis auf triviale Codezeilen der komplette Code in JavaDoc dokumentiert.

¹³<http://www.oracle.com/technetwork/java/codeconv-138413.html>

4.2.4 Begleitende Dokumentation

Durch JavaDoc und die HTML-Dokumentationsdateien kann immer in die Dokumentation eingesehen werden.

4.3 Unterstützende Projektfunktionen

In Abschnitt 2.4 werden unsere Phasenleiter benannt, die für die jeweiligen Phasen verantwortlich sind und als Ansprechpartner und Leiter dienen.

Unsere Projektdateien stehen jederzeit auf *GitHub*¹⁴ zur Verfügung. Außerdem werden regelmäßig von *jedem* Gruppenmitglied Datensicherungen, in Form von Updates und Backups des Repositorys auf dem eigenen Rechner stattfinden.

5 Arbeitspakete, Zeitplan und Budget (Tobias u. Tim)

5.0.1 Annahmen

Wir gehen von folgenden Annahmen aus:

- jedes Mitglied der Gruppe hat rund 14,5 Stunden Zeit in der Woche für das Projekt, davon ausgehend, dass die Veranstaltung SWP2 9CP bringt, was 270 Stunden an Zeit entspricht (1CP = 30 Stunden). Diesen Wert aufgesplittet in 19 Wochen ergibt rechnerisch 14,21 Stunden, also aufgerundet 14,5 Stunden pro Mitglied.
- eine Arbeitswoche hat 5 Tage, wobei jeder Arbeitstag 8 Arbeitsstunden hat.
- des Weiteren ist von folgenden Feiertagen und Urlaubszeiten auszugehen:
 - Weihnachten: 25. und 26. Dezember 2013
 - Neujahr: 01. Januar 2014
 - Ferien/Urlaub: 24. Dezember bis 31. Dezember 2013

5.0.2 Anmerkungen

Hinweis: *Wir haben die Punkte 'Arbeitspakete', 'Zeitplan und Abhängigkeiten' sowie 'Ressourcenanforderung' zu einem Unterpunkt zusammengefasst.*

Wir haben bisher nur die Phasen 'Projektplan' und 'Anforderungsspezifikation' vollständig in den Arbeitspaketen und Zuteilungen behandelt. Die übrigen Phasen des Projekts - Entwurf, Implementierung und Test - können zu diesem Zeitpunkt noch

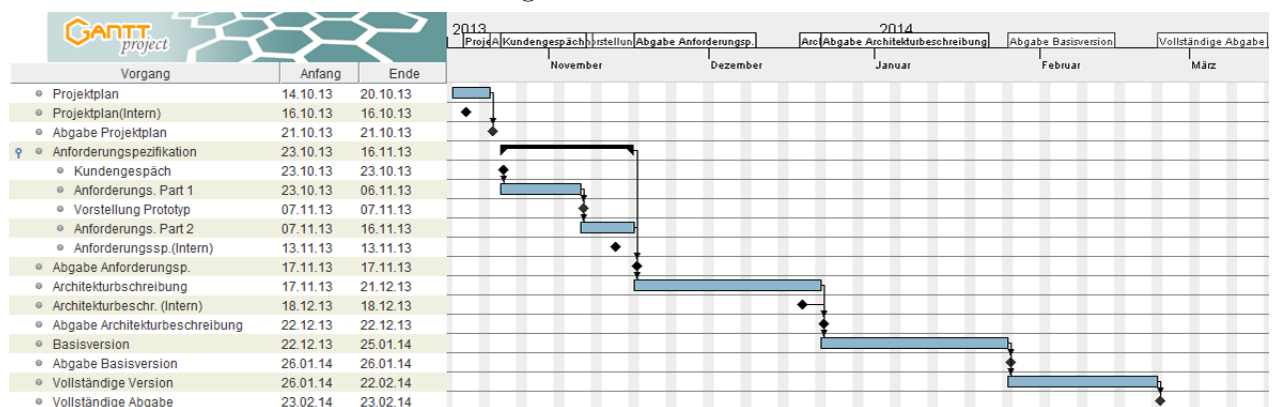
¹⁴<http://github.com/PatMosby/R3VOLUTION>

nicht detailliert beschrieben werden, sondern erfolgen stattdessen in grobem Format.

5.1 Arbeitspakete, Zeitplan, Abhängigkeiten u. Ressourcenanforderungen

Im Folgenden liefert eine grafische Übersicht in Form eines Gantt-Diagramms die grundlegenden Arbeitspakete (Abbildung 1).

Abbildung 1: Gantt-Ueberblick

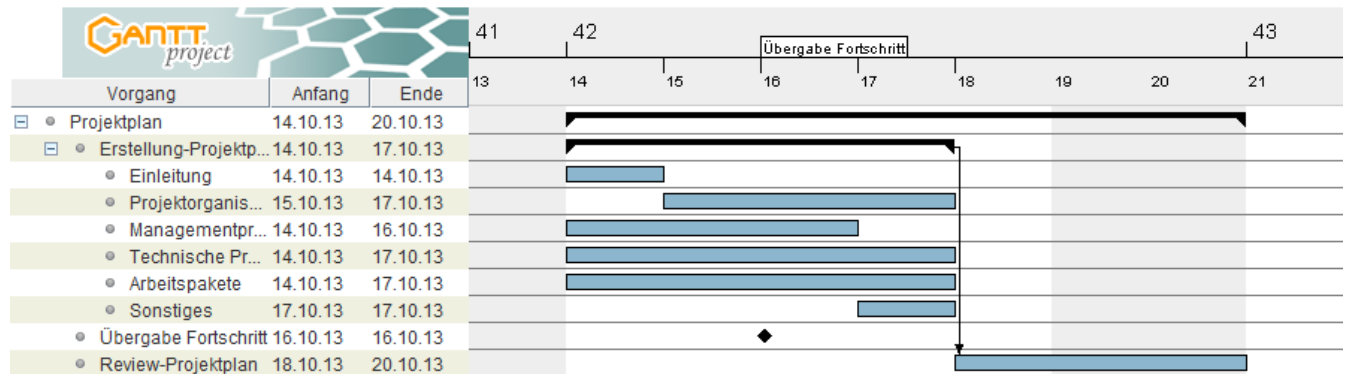


Hinweise zu den Begriffen in den Tabellen: Der Punkt 'Gesamtdauer' der jeweiligen Arbeitspakete ist der Zeitbereich zwischen dem Beginn und dem Ende einer bestimmten Aktivität. Der 'Aufwand' ist die tatsächlich aufgewendete Zeit, in der auch gearbeitet wurde. Somit kann die 'Gesamtdauer' häufig höher ausfallen, wenn sich bestimmte Aktivitäten z.B. über Feiertage hinziehen. Mit der 'Abhängigkeit' ist gemeint, ob das Arbeitspaket von anderen abhängig ist, also im Prinzip einen Vorgänger hat. Mit den 'Ressourcen' sind stets Akteure unserer Gruppe gemeint. Der Punkt 'Mindestanforderungen' beschreibt gewissermaßen das Minimalziel des Arbeitspakets (Minimalbedingung), die mindestens erfüllt werden muss, damit es fertig ist.

5.1.1 Projektplan

In den folgenden Tabellen sind die Arbeitspakete des Abschnitts 'Projektplan' dargestellt. Grafisch sind diese in einem weiteren Gantt-Diagramm realisiert (Abbildung 2).

Abbildung 2: Gantt-Projektplan



A R B E I T S P A K E T 1

Bezeichnung	Gesamter Projektplan
Beschreibung:	Anfertigung und Komplettierung des initialen Projektplans. Details erfolgen in den jeweiligen Unterpunkten.
Hauptverantwortlicher	Tobias Dellert
Abhängigkeit	-
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	14.10.2013
Ende	20.10.2013
Mindestanforderungen:	Die Fertigstellung des initialen Projektplans ist erfolgt und wird durch das interne Gruppen-Review von allen Gruppenmitgliedern überprüft und abgesegnet (Qualitätssicherung).

A R B E I T S P A K E T 1.1

Bezeichnung	Projektplanerstellung
Beschreibung: siehe Unterpunkte	
Hauptverantwortlicher	Tobias Dellert
Abhängigkeit	-
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	16.10.2013
Ende	17.10.2013
Mindestanforderungen: Die Unterpunkte des Projektplans wurden abgearbeitet und der Projektplan kann im Gruppenreview überprüft werden.	

A R B E I T S P A K E T 1.1.1

Bezeichnung	Einleitung
Beschreibung: Erledigung des Projektplanteils	
Hauptverantwortlicher	Patrick Damrow
Abhängigkeit	-
Ressourcen	<ul style="list-style-type: none"> • Patrick Damrow
Aufwand, Gesamtdauer	
Beginn	16.10.2013
Ende	17.10.2013
Mindestanforderungen: Sobald der Projektplanteil fertig ist, kann er zum Gruppenreview weitergegeben werden.	

A R B E I T S P A K E T 1.1.2

Bezeichnung	Projektorganisation
Beschreibung: Erledigung des Projektplantails	
Hauptverantwortlicher	Patrick Damrow
Abhängigkeit	-
Ressourcen	<ul style="list-style-type: none"> • Patrick Damrow
Aufwand, Gesamtdauer	
Beginn	16.10.2013
Ende	16.10.2013
Mindestanforderungen: Sobald der Projektplantail fertig ist, kann er zum Gruppenreview weitergegeben werden.	

A R B E I T S P A K E T 1.1.3

Bezeichnung	Managementprozess
Beschreibung: Erledigung des Projektplantails	
Hauptverantwortlicher	Daniel Pupat
Abhängigkeit	-
Ressourcen	<ul style="list-style-type: none"> • Daniel Pupat
Aufwand, Gesamtdauer	
Beginn	16.10.2013
Ende	16.10.2013
Mindestanforderungen: Sobald der Projektplantail fertig ist, kann er zum Gruppenreview weitergegeben werden.	

A R B E I T S P A K E T 1.1.4

Bezeichnung	Technische Prozesse
Beschreibung: Erledigung des Projektplantails	
Hauptverantwortlicher	Sebastian Bredehöft
Abhängigkeit	-
Ressourcen	<ul style="list-style-type: none"> • Sebastian Bredehöft
Aufwand, Gesamtdauer	
Beginn	16.10.2013
Ende	16.10.2013
Mindestanforderungen: Sobald der Projektplantail fertig ist, kann er zum Gruppenreview weitergegeben werden.	

A R B E I T S P A K E T 1.1.5

Bezeichnung	Arbeitspakete, Zeitplan u. Budget
Beschreibung: Erledigung des Projektplantails	
Hauptverantwortlicher	Tobias Dellert, Tim Ellhoff
Abhängigkeit	-
Ressourcen	<ul style="list-style-type: none"> • Tobias Dellert • Tim Ellhoff
Aufwand, Gesamtdauer	
Beginn	16.10.2013
Ende	17.10.2013
Mindestanforderungen: Sobald der Projektplantail fertig ist, kann er zum Gruppenreview weitergegeben werden.	

A R B E I T S P A K E T 1.1.6

Bezeichnung	Sonstige Elemente
Beschreibung: Erledigung des Projektplantails	
Hauptverantwortlicher	Mohamadreza (Amir) Khostevan
Abhängigkeit	-
Ressourcen	<ul style="list-style-type: none"> • Mohamadreza (Amir) Khostevan
Aufwand, Gesamtdauer	
Beginn	16.10.2013
Ende	16.10.2013
Mindestanforderungen: Sobald der Projektplantail fertig ist, kann er zum Gruppenreview weitergegeben werden.	

A R B E I T S P A K E T 1.2

Bezeichnung	Gruppenreview des Projektplans
Beschreibung: Im internen Gruppenreview wird der fertige Projektplan diskutiert, ggf. verbessert bzw. noch geändert.	
Hauptverantwortlicher	Tobias Dellert, Tim Ellhoff
Abhängigkeit	1 Projektplan, 1.1 Projektplanerstellung
Ressourcen	<ul style="list-style-type: none"> • Tobias Dellert • Tim Ellhoff • Patrick Damrow • Sebastian Bredehöft • Daniel Pupat • Mohamadreza (Amir) Khostevan
Aufwand, Gesamtdauer	
Beginn	19.10.2013
Ende	19.10.2013
Mindestanforderungen: Der Projektplan ist nun von allen Gruppenmitgliedern in einem Review noch einmal auf den Prüfstand gestellt worden und ist nun fertig.	

5.1.2 Anforderungsspezifikation

In den folgenden Tabellen sind die Arbeitspakete des Abschnitts 'Anforderungsspezifikation' dargestellt. Grafisch sind diese in einem weiteren Gantt-Diagramm realisiert (**TODO: Gantt-Diagramm+ Referenz**).

<i>A R B E I T S P A K E T 2</i>	
Bezeichnung	Anforderungsspezifikation
Beschreibung: siehe Unterpunkte	
Hauptverantwortlicher	Patrick Damrow
Abhängigkeit	-
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	21.10.2013
Ende	17.11.2013
Mindestanforderungen: Die Anforderungsspezifikation wird nach dem Abgleich mit dem Kunden komplett fertiggestellt.	

<i>A R B E I T S P A K E T 2.1</i>	
Bezeichnung	Ist-Analyse u. Kundeninterview
Beschreibung: siehe Unterpunkte	
Hauptverantwortlicher	Patrick Domrow
Abhängigkeit	-
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	16.10.2013
Ende	31.10.2013
Mindestanforderungen: Analyse und Verschriftlichung des Ist-Zustands anhand eines Kundeninterviews.	

A R B E I T S P A K E T 2.1.1

Bezeichnung	Kundeninterview
Beschreibung:	
Kundentermin in der Oberschule Rockwinkel mit zwei Mitgliedern aus der Gruppe. Einer führt das Interview, der andere protokolliert. Probleme oder Unstimmigkeiten früh ansprechen und Lösungen finden.	
Hauptverantwortlicher	Tim Ellhoff
Abhängigkeit	-
Ressourcen	<ul style="list-style-type: none"> • Tim Ellhoff • Tobias Dellert • Vertreter der Oberschule Rockwinkel (Kunde)
Aufwand, Gesamtdauer	
Beginn	31.10.2013
Ende	31.10.2013
Mindestanforderungen:	
Erkenntnis darüber, was sich der Kunde genau wünscht und Festhaltung in schriftlicher Form.	

A R B E I T S P A K E T 2.1.2

Bezeichnung	Auswertung des Interviews
Beschreibung:	
Auswertung des Interviews anhand des Protokolls bzw. der Aussagen des Kunden.	
Hauptverantwortlicher	Tim Ellhoff
Abhängigkeit	2.1.1 Kundeninterview
Ressourcen	<ul style="list-style-type: none"> • Tim Ellhoff • Tobias Dellert
Aufwand, Gesamtdauer	
Beginn	02.11.2013
Ende	02.11.2013
Mindestanforderungen:	
Ziel ist eine präzise Auswertung aller Fakten und Erkenntnisse des Kundeninterviews in schriftlicher Form.	

A R B E I T S P A K E T 2.1.3

Bezeichnung	Analyse ähnlicher Systeme
Beschreibung:	Analyse bereits bestehender Bibliotheks-Softwaresysteme. Augenmerk auf etablierte Standards, oder Spezifika
Hauptverantwortlicher	Sebastian Bredehöft
Abhängigkeit	-
Ressourcen	<ul style="list-style-type: none"> • Sebastian Bredehöft • Mohamadreza (Amir) Khostevan • Daniel Pupat
Aufwand, Gesamtdauer	
Beginn	05.12.2013
Ende	06.11.2013
Mindestanforderungen:	Ziel ist es, ein oder mehrere bereits bestehende Systeme zu analysieren und die Erkenntnisse darüber zu verschriftlichen.

A R B E I T S P A K E T 2.1.4

Bezeichnung	Auswertung und Erfassung des Ist-Zustands
Beschreibung:	Sämtliche Erkenntnisse aus den Analysen des Interviews sowie aus den Analysen ähnlicher Systeme werden schriftlich ausgewertet und festgehalten.
Hauptverantwortlicher	Sebastian Bredehöft
Abhängigkeit	2.1.2 Auswertung des Interviews u. 2.1.3 Analyse ähnlicher Systeme
Ressourcen	<ul style="list-style-type: none"> • Sebastian Bredehöft • Mohamadreza (Amir) Khostevan
Aufwand, Gesamtdauer	
Beginn	06.11.2013
Ende	06.11.2013
Mindestanforderungen:	Ziel ist, dass der Ist-Zustand anhand der Analysen genau, detailliert und präzise beschrieben ist.

<i>A R B E I T S P A K E T 2.2</i>	
Bezeichnung	Soll-Zustand
Beschreibung: siehe Unterpunkte	
Hauptverantwortlicher	Daniel Papat
Abhängigkeit	2.1.2 Auswertung des Interviews
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	20.10.2013
Ende	07.11.2013
Mindestanforderungen: Aus vielen Analysen und Erkenntnissen ist die Grundlage für die Anforderungsspezifikation gereift, die erstellt wird.	

5.1.3 Architekturbeschreibung

In den folgenden Tabellen sind die Arbeitspakete des Abschnitts 'Architekturbeschreibung' dargestellt. Grafisch sind diese in einem weiteren Gantt-Diagramm realisiert(**TODO: Gantt-Diagramm+ Referenz**).

5.1.4 Implementierung

In den folgenden Tabellen sind die Arbeitspakete des Abschnitts 'Implementierung' dargestellt. Grafisch sind diese in einem weiteren Gantt-Diagramm realisiert(**TODO: Gantt-Diagramm+ Referenz**).

5.1.5 Test

In den folgenden Tabellen sind die Arbeitspakete des Abschnitts 'Test' dargestellt. Grafisch sind diese in einem weiteren Gantt-Diagramm realisiert(**TODO: Gantt-Diagramm+ Referenz**).

5.1.6 Dokumentabgaben

5.1.7 Sonstiges

5.1.8 Meetings

5.1.9 Kritischer Pfad

6 Sonstige Elemente (Amir)

ENTFÄLLT

6.1 Pläne für die Konvertierung von Daten

ENTFÄLLT

6.2 Managementpläne für Unterauftragsnehmer

ENTFÄLLT *Wenn Fremdbibliotheken benutzt werden...*

6.3 Ausbildungspläne

ENTFÄLLT *Hierunter fallen z.B. auch interne Schulungen, die Ihr durchführen wollt.*

6.4 Raumpläne

ENTFÄLLT ...

6.5 Installationspläne

ENTFÄLLT ...

6.6 Pläne für die Übergabe des Systems

ENTFÄLLT ...